UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/734,690	12/12/2003	Roch Archambault	YOR920030412US1	1217

34663 7590 07/19/2007

MICHAEL J. BUCHENHORNER
8540 S.W. 83 STREET
MIAMI, FL 33143

EXAMINER	
GU, SHAWN X	
ART UNIT	**PAPER NUMBER**
2189	

MAIL DATE	DELIVERY MODE
07/19/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

mN

Office Action Summary	Application No.	Applicant(s)
	10/734,690	ARCHAMBAULT ET AL.
	Examiner	Art Unit
	Shawn X. Gu	2189

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 July 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,4-11 and 14-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,4-11 and 14-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. This Office action is in response to the amendment filed on 2 July 2007. Claims 1, 4-11 and 14-31 are pending. Claims 2, 3, 12 and 13 are cancelled. All objections and rejections not repeated below are withdrawn.

Claim Objections

2. Claims 4, 11 and 14-23 are objected to because of the following informalities:

Per claim 4, the claim is incorrectly indicated as cancelled.

Per claim 11, on the fourth to last line, the word "thread" should be followed by a comma or a semicolon.

All dependent claims are objected to for having the same deficiencies contained in the claims they are dependent from. Appropriate correction is required.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 11 and 14-23 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Per claim 11, on the last two lines, the limitations “the control data structure” and “the pointer” lack sufficient antecedent basis.

All dependent claims are rejected for having the same deficiencies contained in the claims they are dependent from. Appropriate correction is required.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claim 11, 14-25 and 29-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Long et al. [US 2002/0129079 A1] (hereinafter “Long”), and further in view of Tremblay et al. [US 2001/0042188 A1] (hereinafter “Tremblay”) and Applicant Admitted Prior Art (hereinafter “AAPA”).

Per claims 11 and 24, Long teaches a runtime system (see Fig 7) comprising:

a plurality of processors (CPUs 1032, see Fig 6, and Pg. 5, Para. [0056]);

a shared data directory (combination of Pool 110 of Shared Object 108 and Shared Freelist 112, see Fig 1 and Pg. 1, Para. [0009]), for locating and managing shared objects (the shared Freelist 112 is either by itself or in combination with other

component used for locating and managing the shared objects 108 and the shared monitors 106), the directory maintaing shared data entries (Monitors 116, see Figs. 1, 2a-2b, 3a-3c) related to shared data structures (Shared Objects 108, see Figs. 1, 2a-2b, 3a-3c) that are shared by more than one of the plurality of threads; and

control structures (combination of Garbage Collector, the methods and apparatus to associate monitors and objects, and memory management in Java and Titanium, see Pg. 2, Para. [0014] and [0020], Pg. 3, Para. [0040]-[0042], and Figs. 1, 2a-2b, 3a-3c, 4, 5a-5c) to access, allocate and de-allocate the shared data structures through the shared data directory.

Long further teaches the runtime systems operates as a shared memory machine (see Fig. 6, RAM 1034 must be shared by CPUs 1032, also see page 5, paragraph [0056])

Long does not teach "a private memory of each thread, ... said directory is replicated across all of the threads in a private memory of each thread."

Tremblay further teaches a private memory of each thread, the private memory comprising a replica of shared variables such that the variables are replicated across all of the threads (see Tremblay, page 1, para. [0008]), so that the threads can concurrently access and modify the shared variables while maintaining consistency and coherency (see Tremblay, page 1, para. [0007]-[0008]). It is also clear that concurrent accesses to shared data provide higher throughput and performance speed than serial accesses. Therefore, it would have been obvious to one ordinarily skilled in the art at

the time of the Applicant's invention to combine Tremblay with Long to increase throughput and speed while maintaining coherency.

Long further teaches a programming language (Java, see Pg. 5, Para. [0058] and Pg. 6, Para. [0062]), having a plurality of threads (Threads 1-N 104, see Fig 1 and Pg. 1, Para. [0009]) that access memory in a global address space system.

The combined teaching of Long and Tremblay does not teach "a global address space language program ... in a global address space system". However, AAPA teaches global address space (GAS) languages provide a shared memory programming modem abstraction that can be implemented on machines that do not provide shared memory, and an example of GAS languages is Titanium (see AAPA, Background of the Invention). Long's invention involves Java (see Long, page 5, paragraph [0058] and page 6, paragraph [0062]) and it is clear to one ordinarily skilled in the art that Titanium is an extension of Java. Therefore, it would have been obvious to one ordinarily skilled in the art to use Titanium instead of Java in Long's invention in order to provided shared memory programming modem abstraction that can be implemented on machines that do not provide shared memory.

Long further teaches a calling thread (any of the above said threads can be a calling thread) allocates and inserts a handle (the object the thread attempts to acquire/lock, see page 1, paragraph [0009]) in its partition in a common directory of shared variables (Pool 110 of shared objects 108 and Shared Freelist 112) and any thread can directly access the control data structure and the pointer.

It is clear the method of claim 24 is performed by the runtime system of claim 11.

Per claim 14, Long further teaches each of the shared data structures has affinity to a thread that called the allocation or de-allocation routine (arrays in Java are created dynamically, and a thread type class object that created the array obtains the monitor to access the array).

Per claim 15, Long further teaches the shared data structures comprise shared scalar variables (provided by Java), objects (Objects, see Fig. 2a-2b, Fig. 3a-3b), arrays (provided by Java) or pointers (provided by Java).

Per claim 16, Long further teaches that a shared scalar variable is accessed by dereferencing a shared data directory partition for which the shared scalar variable has affinity (Object Pointers 310 to shared objects that are Java scalar variable type objects, see Fig. 3a-3b).

Per claim 17, Long further teaches a shared array has a shared data directory partition that points to a control structure that points to the shared array (Java array type objects are accessed through pointers, which are part of the control structure described above in claim 11).

Per claim 18, Long further teaches the runtime system allocates a controller harness for a shared pointer when the shared pointer is declared by allocating a shared

control block and a shared address structure (monitors implemented in Java for pointer type objects in the shared directory described above.

Per claim 19, Long further teaches some of the shared pointers have shared targets and some of the shared pointers have private targets (the targets are the monitors, which maybe global (shared) or thread-based (private), see Pg. 1, Para. [0009]).

Per claims 20 and 30, Long further teaches entries to the shared data directory are allocated by an owning thread or, in a synchronized manner by all threads at the same time (monitors implemented in Java handle allocation of objects shared by threads, see Pg. 2, Para. [0014], Pg. 3, Para. [0040]-[0042], Pg.4, Para.[0042]-[0047]), and the owning/calling thread inserts a handle in a partition in the directory of shared variables (a thread acquires/sets lock of monitor for the shared object, see Pg. 2, Para. [0014], Pg. 3, Para. [0040]-[0042], Pg.4, Para.[0042]-[0047]).

Per claim 21, Long further teaches the handle comprises a partition index (Monitor Pointer 314, see Figs. 3a-3c) and a variable index (Object Pointer 310, see Figs. 3a-3b).

Per claim 22, Long further teaches the shared data directory includes a partition that is used to access all statically declared non-scalar variables (the group of monitors

that are all used to handle sharing of objects that contain Java static class variables which are non-scalar).

Per claim 23, Long further teaches each thread has exclusive write access rights to a partition and uses a mutually exclusive partition of the shared data directory (monitors that are associated with objects that are locked by a particular thread is mutually exclusive to other threads, see Pg. 3, Para. [0040]-[0042], Pg.4, Para.[0042]-[0047]).

Per claim 25, Long furthers teaches creating control structures comprises creating a plurality of control structures wherein each control structure controls the allocation and de-allocation of a particular type of shared data structure (in Java and other object oriented programming languages, memory allocation and de-allocation of different types of objects are implemented differently, since different types of objects use memory differently).

Per claim 29, Long further teaches the runtime system is implemented on a shared memory system (see claim 11's rejection set forth above).

Per claim 31, Long further teaches the control structures are common such that any thread can access the common control structures (a shared memory machine implies common control structure, as Java's compiled byte codes and the runtime

environment must be present in the shared memory, see Fig 7; also the combination of Garbage Collector, the methods and apparatus to associate monitors and objects, and memory management in Java and Titanium is accessible to any thread).

7. Claim 1, 4-10 and 26-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Long et al. [US 2002/0129079 A1] (hereinafter "Long"), and further in view of Tanenbaum et al. [Distributed Systems: Principles and Paradigms] (hereinafter "Tanenbaum") and Tremblay et al. [US 2001/0042188 A1] (hereinafter "Tremblay") and Applicant Admitted Prior Art (hereinafter "AAPA").

Per claim 1, the claim is already substantially taught by claims 11 and 24 as described above. Long does not teach the runtime system is implemented on a distributed memory system.

Tanenbaum teaches that a distributed memory system (see Tanenbaum, Pg. 16, Fig. 1-6, Private Memory) provides enhanced fault-tolerance, scalability, throughput and increased storage and processing capabilities of the processing system (Reasons for Replication, see Tanenbaum, Pg. 292-293). Tanenbaum further teaches that full-replication of shared data provides further fault-tolerance (full replication ensures that as long as one copy is still available, operations on the shared data can still be performed, see Pg.292-293). Therefore, it would have been obvious to one ordinarily skilled in the art at the time of the Applicant's invention to combine the teachings of Long and

Tanenbaum to provide enhanced fault-tolerance, scalability and throughput. The combined teachings also suggest full replication of shared data.

Per claims 26 and 27, the claims are already substantially described in the rejections of claims 1, 11 and 24 as set forth above.

Per claim 4, Long further teaches the runtime system is implemented on a shared memory system and the directory of shared variables is stored in a shared memory shared by all threads (Primary Storage/RAM 1034, see Fig. 6 and Pg.5, Para. [0055]; also see claim 11's rejection set forth above). This limitation is also taught by Tremblay in page 1, paragraphs 7-8 (see "master copies of variables from main memory").

Per claim 5, Long further teaches the allocation and de-allocation routines are used for both statically and dynamically allocated data (static class variables in Java and dynamic objects such as arrays are all allocated and de-allocated using the control structure described above).

Per claim 6, Long further teaches arrays that are dynamically allocated have affinity to a thread that called the allocation or de-allocation routine (arrays in Java are created dynamically, and a thread type class object that created the array obtains the monitor to access the array).

Per claims 7 and 8, Long further teaches every thread has a handle for each shared variable that it accesses, and the entries in the directory of shared variables area accessed using the handle (Java threads have handling methods to access objects, which include Object Pointers 310 and Monitor Pointers 314, see Figs. 3a-3c).

Per claim 9, Long further teaches the handle comprises a partition index (Monitor Pointer 314, see Figs. 3a-3c) and a variable index (Object Pointer 310, see Figs. 3a-3b).

Per claim 10, Long further teaches each thread has exclusive write access rights to a partition and uses a mutually exclusive partition of the shared data directory (monitors that are associated with objects that are locked by a particular thread is mutually exclusive to other threads, see Pg. 3, Para. [0040]-[0042], Pg.4, Para.[0042]-[0047]).

Per claim 28, Long further teaches each thread has a private data control structure (Lock, Wait, and Unlock, see Fig 3a-3c) with a pointer (Pointer 310 and Point 314, see Fig 3a-3c) to a shared memory fraction.

Response to Arguments

8. Applicant's arguments with respect to claims 1, 4-11 and 14-31have been fully considered but they are not persuasive. The newly added limitations are taught by Long, in further view of Tremblay, Tanenbaum and Applicant Admitted Prior Art as set forth above.

The Applicant's argument that "[t]here is absolutely no teaching or suggestion in Tanenbaum to store a directory of shared variables in a private memory of each thread across a distributed or shared system such that the directory is replicated across all of the threads" on page 10 of the remarks is respectfully traversed. The Applicant is respectfully reminded that the above limitations are taught by Long in combination with Tanenbaum and Tremblay, not by the combination of Long with Tanenbaum alone. Any argument related to the above limitation should be directed to the teachings of Long, Tanenbaum and Tremblay.

The Applicant also referred repeated to a non-final Office action in the arguments, and is hereby respectfully reminded that the arguments should be directed to the most recent Office action, which is the final Office action mailed on 2 April 2007.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shawn Gu whose telephone number is (571) 272-0703. The examiner can normally be reached on 9am-5pm, Monday through Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Reginald Bragdon can be reached on (571) 272-4204. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Reginald D. Bragdon
REGINALD BRAGDON
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Shawn X Gu
Patent Examiner
Art Unit 2189

14 July 2007